UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/844,993	04/27/2001	Jacob Dreyband	033144-017	1373

23562 7590 01/15/2004

BAKER & MCKENZIE
PATENT DEPARTMENT
2001 ROSS AVENUE
SUITE 2300
DALLAS, TX 75201

EXAMINER	
CHANNAVAJJALA, SRIRAMA T	
ART UNIT	**PAPER NUMBER**
2177	

DATE MAILED: 01/15/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/844,993	DREYBAND ET AL.	
	Examiner	Art Unit	
	Srirama Channavajjala	2177	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 November 2003*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-36* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-36* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Response to Amendment

1. Examiner acknowledges applicants' amendment filed on 11/7/2003, paper no. 9.

2. Claims 1-6,9,11,15-20,23,25,27,29-35 have been amended, paper no. # 9.

3. Claims 1-36 are presented for examination.

Drawings

4. The formal drawings filed on 11/7/2003, paper no. # 10 are subjected to approval by the Draftsperson under 37 CFR 1.84 or 1.152.

Information Disclosure Statement

5. The information disclosure statement filed on 4/27/2001, paper no. # 2 has been considered and a copy was enclosed with this office action, paper no. # 5.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-3,15-17,29-31are rejected under 35 U.S.C. 102(a & b) as being anticipated by Grady et al., UML for XML schema mapping specification published on 12/8/1999, page 1-8.

7. As to Claims 1,15,29, Grady et al., teaches a system which including 'mapping a descriptive language including a data description having a structure complexity into an object oriented data presentation' [see Abstract, page 2, 1.1], Grady is directed to standard object oriented language schemas, more specifically UML for XML schema mapping as detailed in Abstract, further Grady also suggests for example object management group where UML has been established certain standards, as best understood by the examiner, descriptive language is to enhance future extensibility and reusability of information in any embedded system for example XML is one of the suitable tool as detailed in Abstract, 'receiving the data description' [page 2, item 2], Grady specifically directed to mapping data types in XML schema to classes, further Grady teaches data types semantics that are related to XML schema concept, see table in page 3, 'identifying a complex-type element in the data description' [page 3, item 1.4,page 6, item 1.8], identifying complex-type element is integral part in the XML document instances of Grady because firstly Grady is directed to XML schema,[see page 6, item 1.8], secondly, Grady specifically teaches for example defining two different data type(s) as detailed in page 3, item 1.4, further it is noted that complex types in XML schemas are user defined data types that can include other elements or attributes, complex types can contain elements defined as either simple or complex, complex types can also include attributes and groups, whereas simple types can only contain facets [see page 6, item:1.8], As best understood by the examiner, complex types are defined using the complex type element and typically contain combination of element, attribute, and group declaration, as well as references to globally declared

elements and groups, further a complex type can be thought of as a mini-schema that defines the valid structure and data contained within a specific element as detailed in page 6, item 1.8;

'creating an executable object oriented class corresponding to the identified complex-type element, wherein the class includes an internal static class wherein the internal static class corresponds to the structure complexity of the data description' [page 4, 1'5, page 6 example the XML schema], as best understood by the examiner, static class analysis is a kind of data flow analysis that computes a set of classes for each variable and expression in a method is integral part of object oriented language such as C++, further it is noted that descriptive language is not only enhance future extensibility but also reusability of classes and methods, for example a simple **static class** is created as shown below and this is common knowledge object oriented environment.

```
[code]
#include <iostream>
class test {
static int counter;
public:
int getcount() { return counter;}
test();
};
int test::counter = 0;

test::test() {
counter++;
}

int main(void) {
```

```
test xyz, bar;

cout << xyz.getcount() << "\n";
}
[/code].
```

As best understood by the examiner, creating an executable object oriented class is integral part of Grady's teaching because firstly, Grady is directed to UML or Unified Modeling Language is itself a standard object-oriented design language [see ABSTRACT], secondly, executable UML is the next layer of abstraction depends on profile of UML, also describes the data and the behavior, further executable UML doesn't make coding decisions, and make use of compiler to generate code, thirdly executable UML is a subset of UML proper as detailed above, on the other hand, to be useful, this subset of UML must have a way to specify actions at a higher level of abstraction than a programming language. There are many executable profiles that could be defined to select a set of meaningful components and how they interact during execution. For example, a object can invoke methods of other objects, which in turn invoke more methods is part of Grady's Unified modeling language

8. Claims 8,14,22,28,36 have been rejection in the analysis of above Claims and are rejection on that basis.

9. As to Claims 2,16,30 Grady teaches a system which including 'receiving the data description comprises receiving an XML Schema [see page 6, 1.8 XML schema]. As best understood by the examiner, the purpose of XML schema is to define the building

blocks of an DML document, just like a data type definition, further it should be noted fundamental XML schema defines such as: elements that appear in a document, defines attributes that appear, defines which elements are child elements, defines the order of child elements, defines the number of child elements, defines whether an element is empty or can include text, defines data types for elements and attributes, defines default and fixed values for elements and attributes [see Grady: page 4, item 1.5]

10. As to Claims 3,17,31, the limitations of this claim have been noted in the rejection of above claim. In addition, Grady disclosed 'validating the data description ' [see Abstract, page 4 1.5 defining element type]. As best understood by the examiner, XML Schema provides powerful dedicated validation features for things like uniqueness, referential integrity, enumerations, complex types and the various data type facet as suggested by Grady, at page 3, item 1.3.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

11. Claims 4-14,18-28,32-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Grady et al., UML for XML schema mapping specification, 12/08/99 as applied to claims 1,15,29 above, and further in view of Davidson et al., [hereafter Davidson], US Patent No. 6083276.

12. As to Claims 4,10,18,24,32, Grady teaches a system which including XML data description, mapping specification [see Abstract], however, Grady does not specifically teach 'mutator method', although Grady suggests for example standard object oriented

design language that is widely used in software development area. On the other hand, Davidson disclosed 'mutator method' [col 24, line 65-67, col 25, line 1-7], examiner interpreting mutator method corresponds to Davidson's mutator methods as detailed in col 25, line 4-6, fig 5.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Davidson et al., into UML for XML schema mapping specification of Grady et al., because both are directed to mapping the schema [see Grady et al., Abstract, page 2: 1.1; Davidson: fig 1, element 122], both are directed to descriptive language including a data description [see Grady et al. XML example page6; Davidson: col 8, line 10-20, col 21, line 50-65, fig 3A-4A] and both are directed to XML environment and are both from the same field of endeavor. One of ordinary skill in the art at the time of the invention would have been motivated to combine the references because that would have allowed users of Grady's UML for XML schema mapping to control which relative combinations of specific methods, classes satisfies his or her needs as suggested by Davidson et al [col 4, line 48-58].

13. As to Claims 5,11,19,25, and 33, both Grady and Davidson teach 'validity determination as to said data description' [see Grady: Abstract, page 2, 1.1; Davidson: fig 3A-4A], Davidson teaches 'sending request including said data description from a user to a remote server' [fig 1, col 7, line 30-40].

14. As to Claims 6,12,20,26,34, the limitations of this claim have been noted in the rejection of claim above. In addition, Davidson disclosed 'reading said data description into a set of valid descriptor classes' [col 9, line 41-52], 'creating a set of objects out of the data description wherein the occurrence of an object reflects validity' [col 10, line 4-20].

15. As to Claims 7,13,21,27,35, the limitations of this claim have been noted in the rejection of claim above. In addition, Davidson disclosed 'Java, C++,Smalltalk' [col 2, line 21-29].

16. As to Claims 9 and 23, Grady teaches a system which including 'mapping a schema including a structural complexity into an executable object oriented language wherein the object oriented language provides a one to one correspondence between the structural complexity of the Schema and the functionality of the object oriented language' [see Abstract], Grady specifically directed to unified modeling language which is a standard object oriented design language that is used by the object management group, further XML schema is integrated for example developing an object model that represented in DML, describing relationships between XML and system to process it as detailed in page , introduction, Schema corresponds to Grady's XML schema as detailed in page 2, item 1.1.

As best understood by the examiner, an executable object oriented language is integral part of Grady's teaching because firstly, Grady is directed to UML or Unified Modeling Language is itself a standard object-oriented design language [see ABSTRACT], secondly, executable UML is the next layer of abstraction depends on profile of UML, also describes the data and the behavior, further executable UML doesn't make coding decisions, and make use of compiler to generate code, thirdly executable UML is a subset of UML proper as detailed above, on the other hand, to be useful, this subset of UML must have a way to specify actions at a higher level of abstraction than a programming language. There are many executable profiles that could be defined to select a set of meaningful components and how they interact during execution. For example, a object can invoke methods of other objects, which in turn invoke more methods is part of Grady's Unified modeling language;

'receiving said schema' [page 3, item 3, 1.3, section 4, page 6], schema corresponds to XML schema as detailed in section 4, page 6, 'validating said schema' [see Abstract, page 4 1.5 defining element type], 'creating a set of object oriented classes including a set of internal static classes to provide a mapping of the schema into the object oriented language' [page 4, 1'5, page 6 example the XML schema], as best understood by the examiner, static class analysis is a kind of data flow analysis that computes a set of classes for each variable and expression in a method is integral part of object oriented language such as C++, further it is noted that descriptive language is not only enhance future extensibility but also reusability of classes and methods, for

example a simple **static class** is created as shown below and this is common knowledge object oriented environment.

```
[code]
#include <iostream>
class test {
static int counter;
public:
int getcount() { return counter;}
test();
};
int test::counter = 0;

test::test() {
counter++;
}

int main(void) {
test xyz, bar;

cout << xyz.getcount() << "\n";
}
[/code].
```

It is however, noted that Grady does not specifically teach 'instance corresponding to the object oriented classes, compiling the instance', although compiling, instances, creating classes, methods and functions are integral part of any object-oriented language environment. On the other hand, Davidson disclosed 'instance corresponding to the object oriented classes' [col 24, line 51-67], 'compiling the instance', [col 24, line 65-67, col 25, line 1-7].

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Davidson et al., into UML for XML

schema mapping specification of Grady et al., because both are directed to mapping the schema [see Grady et al., Abstract, page 2: 1.1; Davidson: fig 1, element 122], both are directed to descriptive language including a data description [see Grady et al. XML example page6; Davidson: col 8, line 10-20, col 21, line 50-65, fig 3A-4A] and both are directed to XML environment and are both from the same field of endeavor. One of ordinary skill in the art at the time of the invention would have been motivated to combine the references because that would have allowed users of Grady's UML for XML schema mapping to control which relative combinations of specific methods, classes satisfies his or her needs as suggested by Davidson et al [col 4, line 48-58].

Response to Arguments

17. Applicant's arguments filed on 11/7/2003 with respective to Claims 1-36 have been fully considered but they are not persuasive, for examiner's response, see discussion below:

.a) At page 12, Claims 1,15,29, applicant argues "Grady does not teach the claimed process of identifying a complex-type data element in a data description and using that complex-type data element to generate an object oriented class comprising an internal static class.....

As to the above argument [a], examiner disagree with the applicant because firstly, Grady is directed to an object oriented design language, more specifically Unified

modeling language which is a standard object-oriented design language [see Abstract], secondly, Grady teaches a mapping between XML schema and UML as detailed in page 2, item 1.1, thirdly, identifying complex-type element is integral part in the XML document instances of Grady because the following reasons (i) Grady is directed to XML schema,[see page 6, item 1.8], (ii), Grady specifically teaches for example defining two different data type(s) as detailed in page 3, item 1.4, further it is noted that complex types in XML schemas are user defined data types [see page 2, item 1.2] that can include other elements or attributes, complex types can contain elements defined as either simple or complex, complex types can also include attributes and groups, whereas simple types can only contain facets [see page 6, item:1.8], As best understood by the examiner, complex types are defined using the complex type element and typically contain combination of element, attribute, and group declaration, as well as references to globally declared elements and groups, further a complex type can be thought of as a mini-schema that defines the valid structure and data contained within a specific element as detailed in page 6, item 1.8. As best understood by the examiner, static class analysis is a kind of data flow analysis that computes a set of classes for each variable and expression in a method is integral part of object oriented language such as C++, further it is noted that descriptive language is not only enhance future extensibility but also reusability of classes and methods, for example a simple **static class** is created as shown below and this is common knowledge in standard object oriented language such as UML or unified modeling language [see Abstract].

b) At page 12, Claims 1,15,29, Applicant argues, Grady does not disclose or suggest that executable code can be generated based upon a UML map.

As to the above argument [b], examiner disagree with the applicant because executable object oriented class is integral part of Grady's teaching because firstly, Grady is directed to UML or Unified Modeling Language is itself a standard object-oriented design language [see ABSTRACT], secondly, executable UML is the next layer of abstraction depends on profile of UML, also describes the data and the behavior, further executable UML doesn't make coding decisions, and make use of compiler to generate code, thirdly executable UML is a subset of UML proper as detailed above, on the other hand, to be useful, this subset of UML must have a way to specify actions at a higher level of abstraction than a programming language. There are many executable profiles that could be defined to select a set of meaningful components and how they interact during execution. For example, a object can invoke methods of other objects, which in turn invoke more methods is part of Grady's Unified modeling language

Examiner applies above discussed arguments to Claims 4-14,18-28,32-36, depend from independent Claims1, 15,29.

c) At page 14, Claims 9 and 23, applicant agues "Grady, however, does not teach the claimed process of creating a set of object oriented classes including a set of static classes....

As to the above argument [c], examiner disagree with the applicant because Grady specifically teaches Unified Modeling Language which is a standard object-oriented language as detailed in Abstract, it is common knowledge in any object oriented language to have functions, attributes, defining data types, defining class, defining objects and like. As best understood by the examiner, static class analysis is a kind of data flow analysis that computes a set of classes for each variable and expression in a method is integral part of object oriented language such as C++, further it is noted that descriptive language is not only enhance future extensibility but also reusability of classes and methods, for example a simple **static class** is created as shown below and this is common knowledge object oriented environment.

```
[code]
#include <iostream>
class test {
static int counter;
public:
int getcount() { return counter;}
test();
};
int test::counter = 0;

test::test() {
counter++;
}

int main(void) {
test xyz, bar;

count << xyz.getcount() << "\n";
}
[/code].
```

While it is noted that Grady does not specifically teach 'instance corresponding to the object oriented classes, compiling the instance', although compiling, instances, creating classes, methods and functions are integral part of any object-oriented language environment. On the other hand, Davidson disclosed 'instance corresponding to the object oriented classes' [col 24, line 51-67], 'compiling the instance', [col 24, line 65-67, col 25, line 1-7].

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Davidson et al., into UML for XML schema mapping specification of Grady et al., because both are directed to mapping the schema [see Grady et al., Abstract, page 2: 1.1; Davidson: fig 1, element 122], both are directed to descriptive language including a data description [see Grady et al. XMLexample page6; Davidson: col 8, line 10-20, col 21, line 50-65, fig 3A-4A] and both are directed to XML environment and are both from the same field of endeavor. One of ordinary skill in the art at the time of the invention would have been motivated to combine the references because that would have allowed users of Grady's UML for XML schema mapping to control which relative combinations of specific methods, classes satisfies his or her needs as suggested by Davidson et al [col 4, line 48-58].

Conclusion

The prior art made of record

a. Grady et al., UML for XML schema mapping specification published on 12/8/1999, page 1-8.

b. US Patent No. 6083276

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure

c. US Patent No. 5794030

d. US Patent No. 6540142

e. US Patent No. 6569207

f. US Patent No. 6418446

g. US Patent No. 6026408

h. US Patent No. 5797137

i. US Patent No. 6490581

j. US Patent No. 5956730

k. US Patent No. 5809505

l. US Patent No. 6446256

m. Lucian et al., Mapping XML and relational schemas with clio, 2 pages

n. Migrating from XML DTD to XML schema using UML, Rational Software white paper, year 2000, pages 1-8

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Srirama Channavajjala whose telephone number is (703) 308-8538. The examiner can normally be reached on Monday-Friday from 8:00 AM to 5:30 PM Eastern Time. The TC2100's Customer Service number is (703) 306-5631.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John E. Breene, can be reached on (703) 305-9790. The fax phone numbers for the organization where the application or proceeding is assigned are as follows:

703/746-7238	**(After Final Communication)**
703/872-9306	**(Offical Communications)**
703/746-7240	**(For Status inquiries, draft communication)**

Any inquiry of general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-9600.

sc
Patent Examiner.
January 13, 2004.